

Development Areas

This page is obsolete. We now use GitHub issue tickets and projects for all in progress development.



- Current GitHub Project Boards: <https://github.com/orgs/DSpace/projects>
- Backend GitHub Tickets: <https://github.com/DSpace/DSpace/issues>
- Frontend GitHub Tickets: <https://github.com/DSpace/dspace-angular/issues>

If your work on DSpace involved any of the aspects below, please go to the relevant page to see what's going on, and add yourself to the list of interested parties! (Just add your name to the bottom of the page.) The goal of the DSpace Community Process is to establish an organizational process to the creation of new Proposals, Special topic Meetings, and their eventual "graduation" into DSpace Development Projects with scheduled integration timeframes to get into DSpace.

DSpace architecture white paper by [RobertTansley - dspace-architecture-whitepaper.pdf](#)

Legacy Idea Pool

Please work through these idea pages (children of the DevelopmentAreas page) and the rest of the hardcoded list below to consolidate the sections of this page into labels for above grouping.

- [AddOnMechanism](#)
 - [AddOnMechanism 2fAddOnPrototype](#)
 - [TapirAddOn](#)
 - [WikiCourse](#)
- [AipBackupRestoreDiscussion](#)
 - [AipCoreAPIChanges](#)
 - [AIP Export Implementation Notes](#)
 - [DuraCloud DSpace Interaction Notes](#)
- [AipPrototype](#)
 - [AipPrototype RWComments](#)
- [AssetStore](#)
 - [AssetStoreClippings](#)
 - [AssetStoreIssues](#)
 - [AssetStoreUseCases](#)
 - [RobsAssetStorePrototype](#)
- [Audio/Video Streaming \(RTMP, HLS, DASH\) Support](#)
- [AuthorisationSystem](#)
- [Authority Control of Metadata Values](#)
- [Author Profiles XMLUI](#)
- [Basic RDF implementation](#)
- [Batch Metadata Editing Feature](#)
 - [Batch Metadata Editing Prototype](#)
- [BitstreamFormat Renovation](#)
 - [About Data Formats](#)
 - [BitstreamFormat Conversion Instructions](#)
 - [BitstreamFormat Renovation Use Cases](#)
 - [BitstreamFormat Workbench](#)
 - [Talk__BitstreamFormat+Renovation](#)
- [Bitstream order](#)
- [BitstreamRelationships](#)
- [BrowseSystem](#)
- [CAS and SAML Authentication](#)
- [CheckerAddOnTechnique](#)
- [Citation Cover Page](#)
- [Citation Generation](#)
- [CleanupTasks](#)
- [CommonsConfigurationProposal](#)
- [ContextGuidedIngest](#)
- [CrosswalkPlugins](#)
 - [XsltCrosswalk](#)
- [DAO Prototype](#)
 - [DSpace DAO Prototype](#)
- [Debug lucene query](#)
- [Digital Author Identifier](#)
- [Dimensions Integration](#)
- [DocumentationTasks](#)
- [Document Type Based Submission](#)
 - [File__DocumentTypeBasedSubmission.zip](#)
- [Document Viewer Integration](#)
- [DSpace 2.0](#)
 - [DSpace 2.0 AuthNZ](#)
 - [DSpace 2.0 Comparing Existing Technologies](#)
 - [DSpace 2.0 Core Services](#)
 - [DSpace 2.0 Design Guidelines](#)
 - [DSpace 2.0 Expressing DSpace Domain Model In RDF](#)
 - [DSpace 2.0 Getting Started](#)

- DSpace 2.0 Kernel
 - DSpace 2.0 Modelling DSpace Objects
 - DSpace 2.0 Modelling Fedora in RDF
 - DSpace 2.0 Modelling Services
 - DSpace 2.0 Original Proposal
 - DSpace 2.0 Pluggable Storage
 - DSpace 2.0 Requirements and Issues
 - DSpace 2.0 Road Map
- DSpace and Docker
 - DSpace Docker and Cloud Deployment Goals
 - Past DSpace Docker Implementation Conversations
 - Running DSpace 7 with Docker
 - Sample Output Accessing Built Images
- DSpace Backend as One Webapp
- DSpace Elsevier API integration
- DSpace Futures
 - Brainstorms on a Future UI
 - DSpace Futures DSpace and Hydra Meeting Notes
 - DSpace Futures - REST API
 - DSpace Futures REST API Meeting Notes
 - DSpace Futures REST API Use Cases
- DSpaceIntermediateMetadata
- DSpaceMETSAIPProfile
- DSpaceMETSSIPProfile
- DSpace Service based api
 - DSpace Service based api: API Changelist
 - DSpace service based api: Porting DSpace modules
- DSpaceStatistics
 - AdministrativeStatistics
 - DstatPackage
 - ReportGeneration
 - StatisticsAndLog4jIdeas
 - StatisticsFurtherSpeculations
 - StatisticsProposalOne
- DSpace statistics - current status and future development
- DSpace Testing
- DynamicBrowsePrototype
- Editable Authority Control
- Email Deposit
- Embargo 1.6
- Enhanced Configuration Scheme
- EventSystemPrototype
 - EventMechanism
 - SimpleEventHandling
- EvolutionToServiceLocatorProposal
- Fedora Integration
 - DSpace Fedora Collaboration
- FlexPaper Document Viewer for XMLUI
- FlexPaper Integration into XMLUI Brainstorming
- FormatIdentificationIssues
- Healthcheck reports
- Hibernate prototype
- HistorySystemPrototype
 - HistoryDiscussion
 - HistorySystem
- IIIF and DSpace
 - IIIF/DSpace Meeting April 21, 2017
 - IIIF/DSpace Meeting Aug 25, 2017 at 1500 UTC
 - IIIF/DSpace Meeting June 23, 2017 at 15:00 UTC
 - IIIF/DSpace Meeting Nov 3, 2017 at 1500 UTC
- Ingesting Large Bitstreams
- Installer Prototype
 - Dspace Batch Installer for Window
- Integration of Uncheck plagiarism checker into DSPACE platform
- Internationalization Support (I18nSupport)
 - I18n Update Catalan translation of Messages.properties from 1.4.1 to 1.5.1
 - I18n Update Czech translation of Messages.properties from 1.5 to 1.5.1
- ItemBatchUpdate
- JhoveIntegration
- Jython webapp for DSpace
- LinksManagement
- NetworkInterfaces
- OAI Sets Generalisation
- OpenAIRE Literature v.4
- ORCID Integration for JSPUI
- PersistentIdentifiers
- PersistentIdentifiers Requirements
- PledgePolicyPrototype
- Replace Authentication and Authorization code in DSpace with a 3rd-party framework

- [SakaiIntegration](#)
- [Simple Archive Format Packager](#)
- [Solr](#)
- [SortingSearchResults](#)
- [Tagging](#)
- [XmlNamespaces](#)
 - [XmlNamespaces 2fCwspaceImscp](#)

Everything Under this Point needs to get reorganized and labeled. Use appropriate labels so that project proposals and active work areas show up where they should above.

From Scratchpad...

This page is intended for those of you doing DSpace work to provide a link to wiki pages you may have created to support your development work or to document your procedures.

Typically code will be stored in the "[sandbox](#)" area of the source tree.

- [Dynamic Browse Prototype](#) - [RichardJones](#)
- [OAI Sets Generalisation](#) - [RichardJones](#)
- [Persistent Identifiers \(requirements\)](#) - [JamesRutherford](#)
- [DAO Prototype](#) - [JamesRutherford](#)
- [About Data Formats](#) – [Larry Stone](#)
- [BitstreamFormat Renovation](#) – [Larry Stone](#)
- [Embargos](#) - [Elliot Metsger](#)
- [Hiding Collections/Communities](#) - [Elliot Metsger](#)
- [Modular Usage Statistics](#) - [Mark H. Wood](#)
- [Tagging](#) - [christophe_dupriez](#)
- [Simple Archive Format](#) - [Packager](#) - [Peter Dietz](#)

Architecture

Work towards the DSpace 2.0 architecture: [DSpace 2.0 Original Proposal](#)

Storage refactoring

- [AssetStore](#) - standards-based AIP storage layer for easier preservation
- [AipPrototype](#) - (OBSOLETE - [AIP Backup and Restore](#) has replaced this). Please use prototype AIP implementation, above the asset store layer, for use on DSpace 1.4/1.5. Emphasis is on preservation and custody transfer of archived objects.
- [PluggableStorage](#) - allow easier integration of different asset store backends. Prototypes within.
- [DAO Prototype](#) - removes storage layer dependent code from core classes. Should make it easier to port DSpace to other database platforms.

Modularity

- [ModularityMechanism](#) - Rather old musings
- [PluginManager](#) - Simple first steps
- [AddOnMechanism](#) - Defining a mechanism for build-time add-ons

User interface framework

- [UI Framework](#)
- [Internationalization Support \(I18nSupport\)](#) - Internationalisation progress

Other aspects

- [EventMechanism](#) - Adding event mechanism to the data model
- [EventSystemPrototype](#) - prototype implementation (for DSpace 1.5) of an event driver
- [ScalabilityIssues1.4](#)
- [CleanupTasks](#) - General tasks to ensure the architecture/code is robust and consistent
- [PersistentIdentifiers](#) - making DSpace less dependent on the Handle system.
- [BitstreamFormat Renovation](#) - new design and implementation of data format technical metadata for Bitstreams, to better support digital preservation.

Features/functionality

- [DSpaceStatistics](#)
- [PersistentIdentifiers](#) - Making identifiers 'pluggable' (Handles, ARKs...)
- [NetworkInterfaces](#) - e.g. Web Services, WebDAV etc.
- [HistorySystem](#)
- [Item Versioning Support](#) - approaches to supporting multiple versions over time (i.e. editions, as opposed to multiple formats of the same version)

- [BitstreamRelationships](#) - proposal to encode relationships between bitstreams explicitly in the data model.
- [Citation Cover Page](#) - generating a Citation Cover Page to PDF's downloaded from DSpace on-the-fly
- [Citations/references/originality reports](#) - generating citations/references lists for end user, as well as list of sources with similar content and % of similarity

Metadata

- [MetadataSupport](#) for non-Dublin Core schemas
- [XmlNamespaces](#)
- [DSpaceIntermediateMetadata](#) - XML format for metadata tables for XSLT transforms
- [CrosswalkPlugins](#)
- [PackagerPlugins](#)
- [MetadataDiscovery](#)
- Attaching "policy" metadata to objects, SIPs (experimental)
- [DSpaceMETSSIPProfile](#) (for use with the [Packager Plugin](#), via the [Lightweight Network Interface](#) and other package-based applications)
- [DSpaceMETSAIPPProfile](#) (used by [AIP Prototype](#))
- [ItemBatchUpdate](#)

Security

- [AuthorisationSystem](#)
- [StackableAuthenticationMethods](#)

Preservation

- [DigitalPreservationToolsAndStrategies](#)
- [AIP Backup and Restore](#) - Based on the initial [AipPrototype](#) implementation (by Larry Stone), but only details with exporting/importing Archival Information Packages which describe the entire content hierarchy (Community/Collection/Item) within DSpace. Emphasis is on creating a better "backup" and restore solution, where an entire DSpace could be restored from a directory of these AIPs. *[Released in DSpace 1.7.0]*
 - [AipBackupRestoreDiscussion](#) - Discussion about the [AIP Backup and Restore](#) feature released in 1.7

Search

- [SortingSearchResults](#)

Installation / Updates

- [Installer Prototype](#) - Prototype for a new "easy" installer, which will better automate the Installation/Upgrade processes for DSpace.

Big Data

- [Ingesting Large Bitstreams](#) - uploading big data sets via HTTP
-

Other planned/possible work

- Half-developed: Diagnostics servlet. A servlet that goes around and kicks the apps's tyres a bit on startup or on request. At the moment this just checks that the lucene index is present and unlocked. I'm hoping to extend this to checking that the db is present and correct, and maybe to check that the db and filesystem are in synch. No schema changes. [Ojd20](#)
- Largely conceptual, some code snippets exist: Upgraded submission system ([SubmissionSystem](#)) - this is quite a big job so perhaps not for 1.3 but we'll see. [RichardJones](#)
- Conceptual, shouldn't be too hard: Upgraded workflow system ([WorkflowSystem](#)) - not too big a job I don't think. Hardest part will almost certainly be the interface, as the options could be a little convoluted. [RichardJones](#)
- Hierarchical collections as well as communities. Not implemented.